

## Un preconditionador tipo ILU para problemas de Electromagnetismo Computacional

R. BRU<sup>1</sup>, N. MALLA<sup>2</sup>, J. MARÍN<sup>3</sup>

<sup>1</sup> *Instituto de Matemática Multidisciplinar, Universidad Politécnica de Valencia, E-46022 Valencia.  
E-mail: rbru@mat.upv.es.*

<sup>2</sup> *Instituto de Matemática Multidisciplinar, Universidad Politécnica de Valencia, E-46022 Valencia.  
E-mail: namalmar@doctor.upv.es.*

<sup>3</sup> *Instituto de Matemática Multidisciplinar, Universidad Politécnica de Valencia, E-46022 Valencia.  
E-mail: jmarinma@mat.upv.es.*

**Palabras clave:** Electromagnetismo computacional, métodos iterativos, preconditionador, ILU, particionado de grafos

### Resumen

Las diferentes aplicaciones del Electromagnetismo Computacional requieren la resolución de las ecuaciones de Maxwell mediante métodos numéricos. El principal coste computacional de los diferentes métodos reside en la obtención de una solución aproximada de sistemas de ecuaciones lineales. En el caso de métodos integrales estos sistemas se caracterizan por tener una matriz de coeficientes densa, compleja y frecuentemente mal condicionada. En aplicaciones industriales de gran escala su número de incógnitas puede ser del orden de cientos de miles o incluso de millones por lo que para su resolución se emplean métodos iterativos preconditionados basados en subespacios de Krylov [17].

En este trabajo se presenta un preconditionador factorizado tipo ILU [11, 14, 16] basado en técnicas de particionado de grafos [1, 8]. El preconditionador obtenido ofrece un alto grado de paralelismo y los resultados numéricos preliminares muestran que es competitivo frente a otro tipo de estrategias que aparecen en la bibliografía [2].

## 1. Introducción

Dentro de las diferentes aplicaciones de electromagnetismo computacional se incluyen el cálculo del patrón de difracción de haces de ondas electromagnéticas al incidir sobre un cuerpo grande e irregular o el estudio del diagrama de radiación de una antena. La resolución de estos problemas, que son de gran interés para la industria aeronáutica, requieren el uso de nuevos algoritmos y de equipos informáticos de altas prestaciones.

Los fenómenos electromagnéticos se modelan con las ecuaciones de Maxwell [13]. Para resolver numéricamente estas ecuaciones se utilizan métodos basados en ecuaciones diferenciales o métodos basados en su forma integral. En ambos casos, un sistema de ecuaciones lineales debe ser resuelto tras la discretización del conjunto de ecuaciones. En este trabajo los sistemas a resolver

$$Ax = b \tag{1}$$

se obtienen tras aplicar el Método de los Momentos [7, 15] en su forma integral. La matriz  $A$  es densa, no simétrica, con elementos complejos y mal condicionada.

En aplicaciones industriales a gran escala los sistemas pueden llegar a ser del orden de cientos de miles o de millones de incógnitas haciéndose imposible su almacenamiento en memoria. Por lo tanto, sólo se almacena una aproximación dispersa de la matriz del sistema correspondiente a las interacciones del campo cercano. El resto de entradas se calculan en el instante en el que son requeridas. Por este motivo se emplean métodos iterativos preconditionados basados en subespacios de Krylov [17] tales como GMRES [17] o BICGSTAB [19].

La convergencia de los métodos iterativos de Krylov mejoran con el uso de técnicas de preconditionado. Estas consisten en cambiar el sistema original por otro de idéntica solución, de forma que el número de condición de la matriz del nuevo sistema sea menor, o bien que tenga una mejor distribución de los autovalores.

Al preconditionar el sistema por la derecha este quedaría de la siguiente forma:

$$AMy = b, \quad x = My,$$

donde  $M$  es el preconditionador. También es posible el preconditionado por la izquierda [17]. Debido al gran tamaño de los sistemas a resolver es necesario el empleo de preconditionadores que sean paralelizables.

En [3] se comparan un número de preconditionadores. Los experimentos numéricos demuestran que el preconditionador de inversa aproximada basado en la minimización de la norma de Frobenius SPAI [6] es el más efectivo y que los preconditionadores de tipo ILU [17] no convergen o la velocidad de convergencia es mayor.

En este trabajo nosotros estudiamos un preconditionador factorizado de tipo ILU basado en técnicas de particionado de grafos [1, 8]. Los resultados numéricos demuestran que es competitivo frente a SPAI tanto en el cálculo como en la aplicación.

Este artículo se estructura de la siguiente forma. En la sección 2 se presenta el preconditionador ILU propuesto. En la sección 3 se muestran los resultados de los experimentos numéricos con diferentes matrices proporcionadas por EADS-CASA. Finalmente, la sección 4 contiene las conclusiones más relevantes del estudio.

## 2. Precondicionador tipo ILU

En este trabajo nosotros presentamos un preconditionador factorizado tipo ILU basado en técnicas de particionado de grafos con alto grado de paralelismo.

Los algoritmos de particionado de grafos, muy utilizadas hoy en día en entornos paralelos para dividir un problema entre los procesadores disponibles, requieren como parámetro de entrada el grafo de adyacencia asociado a una matriz.

Para una matriz dispersa  $A$  consideramos su grafo de adyacencia  $G = (V, E)$ , donde  $V = \{1, \dots, n\}$  es el conjunto de nodos (o vértices) y  $E$  es el conjunto de ejes  $\langle i, j \rangle$ ,  $i, j \in V$  y  $a_{ij} \neq 0$ . En nuestro caso, debido a que la matriz  $A$  es densa, se trabaja con el grafo asociado a la matriz  $\hat{A}$  correspondiente al campo cercano de  $A$ . A los ejes del grafo asociado a la matriz  $\hat{A}$  se le añade la distancia entre los vertices. Como  $\hat{A}$  tiene una estructura no simétrica se asocia el grafo de adyacencia de  $\hat{A} + \hat{A}^T$ .

El grafo de adyacencia es dividido por metis [8] en  $p$  subgrafos de aproximadamente el mismo tamaño, minimizando el número de cortes en los ejes. Los nodos que están conectados por ejes cortados son movidos de los subgrafos a un conjunto separador. Al numerar los nodos por subgrafos y tomando en último lugar los nodos del conjunto separador, la matriz quedaría reordenada de la siguiente forma:

$$P^T \hat{A} P = \begin{pmatrix} \hat{A}_1 & & & \hat{B}_1 \\ & \hat{A}_2 & & \hat{B}_2 \\ & & \ddots & \vdots \\ & & & \hat{A}_p & \hat{B}_p \\ \hat{C}_1 & \hat{C}_2 & \dots & \hat{C}_p & \hat{A}_S \end{pmatrix}$$

donde  $P$  es una matriz de permutación. Los bloques diagonales  $\hat{A}_1, \dots, \hat{A}_p$  representan las conexiones entre nodos interiores de los subgrafos,  $\hat{B}_i$  y  $\hat{C}_i$  las conexiones entre subgrafos y  $\hat{A}_S$  las conexiones entre nodos del conjunto separador.

Esta matriz se puede descomponer como,

$$P^T \hat{A} P = \begin{pmatrix} \hat{A}_1 & & & \hat{B}_1 \\ & \hat{A}_2 & & \hat{B}_2 \\ & & \ddots & \vdots \\ & & & \hat{A}_p & \hat{B}_p \\ \hat{C}_1 & \hat{C}_2 & \dots & \hat{C}_p & \hat{A}_S \end{pmatrix} = \begin{pmatrix} \hat{L}_1 & & & \\ & \hat{L}_2 & & \\ & & \ddots & \vdots \\ & & & \hat{L}_p & \\ \hat{F}_1 & \hat{F}_2 & \dots & \hat{F}_p & \hat{L}_S \end{pmatrix} \begin{pmatrix} \hat{U}_1 & & & \hat{E}_1 \\ & \hat{U}_2 & & \hat{E}_2 \\ & & \ddots & \vdots \\ & & & \hat{U}_p & \hat{E}_p \\ & \dots & & & \hat{U}_S \end{pmatrix}$$

donde

$$\hat{A}_i = \hat{L}_i \hat{U}_i, \hat{E}_i = \hat{L}_i^{-1} \hat{B}_i, \hat{F}_i = \hat{C}_i \hat{U}_i^{-1}$$

y  $\hat{L}_S, \hat{U}_S$  son los factores triangulares del complemento de Schur de la matriz

$$\hat{S} = \hat{A}_S - \sum_{i=1}^p \hat{C}_i \hat{A}_i^{-1} \hat{B}_i.$$

No es necesario calcular los bloques  $\hat{E}_i$  y  $\hat{F}_i$  explícitamente.

Los elementos de los bloques diagonales  $\hat{A}_1, \dots, \hat{A}_p$  de  $P^T \hat{A} P$  se reordenan para evitar que al realizar las descomposiciones  $\hat{L}_i \hat{U}_i$  y  $\hat{L}_S \hat{U}_S$  se llenen más de lo normal. Para ello se hace uso de métodos como Minimum Degree, Cuthill-McKee,...

En vez de calcular directamente los factores  $\hat{L}_i \hat{U}_i$  y  $\hat{L}_S \hat{U}_S$  se obtiene una aproximación de ellos con una técnica apropiada de tipo ILU, ya que su cálculo exacto requiere un alto coste computacional. Esta aproximación se utiliza como preconditionador.

El cálculo del preconditionador se divide en dos fases. En la primera fase se calculan los factores aproximados  $\hat{L}_i \hat{U}_i$  de los bloques diagonales  $\hat{A}_i$ . En la segunda fase se calcula la aproximación del complemento de Schur  $\hat{L}_S \hat{U}_S$ . Para el cálculo de  $\hat{C}_i \hat{A}_i^{-1} \hat{B}_i$  se resuelven sistemas triangulares con las aproximaciones de  $\hat{L}_i$  y  $\hat{U}_i$ , ya que  $\hat{L}_i \hat{U}_i \approx \hat{A}_i$ .

Esta aproximación de  $P^T \hat{A} P$  se utiliza como preconditionador para el método iterativo GMRES(m) aplicado a

$$(P^T A P)y = P^T b, y = P^T x.$$

La solución del sistema lineal original  $Ax = b$  se obtiene como  $x = Py$ .

### 3. Experimentos numéricos

En esta sección se muestran los resultados obtenidos de los experimentos numéricos llevados a cabo sobre un conjunto de matrices (listadas en la Tabla 1) proporcionadas por la empresa EADS-CASA. Se ha utilizado el método GMRES(m) preconditionado por el preconditionador de la sección 2. Para el particionado de grafos se ha empleado a metis, y para las descomposiciones ILU de los bloques diagonales  $\hat{A}_i$  y del complemento de Schur  $\hat{A}_S$ , la función `luinc()` de Matlab. Se ha considerado que se obtiene convergencia del método iterativo GMRES(m) cuando el residuo inicial se ha reducido por un factor de  $10^{-6}$ . Los experimentos se han realizado en una estación de trabajo con procesador Intel Core 2 Quad y el código ha sido implementado en Matlab. Algunos de los resultados se muestran en las tablas 3-7, las cuales están formadas por nueve columnas.  $p$  indica el número de particiones realizadas por metis en el grafo de la matriz  $\hat{A}$ . *método* define el método utilizado para ordenar los bloques diagonales  $\hat{A}_i$ ; *symamd* implementa el método Minimum Degree para matrices simétricas, *colamd* para no simétricas, *symrcm* Cuthill-McKee y *colperm* reordena las columnas en orden creciente al número de no ceros. *tol* la tolerancia de vaciado para la función `luinc()` de Matlab.  $nnz(\hat{A})$  el número de no ceros de  $\hat{A}$ .  $nnz(\hat{L} + \hat{U})$  el número de no ceros de las descomposiciones ILU de los bloques diagonales  $\hat{A}_i$  y del complemento de Schur  $\hat{A}_S$ .  $\alpha$  es el ratio que viene dado por  $(nnz(\hat{L} + \hat{U})/nnz(\hat{A}))$ . *Tilu* el tiempo de cálculo del preconditionador. *Tsol* el tiempo de cálculo de solución del sistema. Por último, *iter* define el número de iteraciones necesario para que el método GMRES(m) converja.

En la tabla 2 se muestran resultados para el preconditionador SPAI.

En todos los resultados de las tablas 3-7 correspondientes al preconditionador ILU, el tiempo de cálculo del preconditionador siempre es muy inferior al del preconditionador SPAI que es del orden de  $10^3$ . Para el caso de los resultados de la tabla 3, el tiempo de solución del sistema, el número de iteraciones y la densidad del preconditionador de tipo ILU también es inferior a los del preconditionador SPAI en todos los casos. Esto también sucede en un resultado de la tabla 4 para el método de reordenación *symrcm*.

Al comparar el número de no ceros ( $nnz(\hat{L} + \hat{U})$ ) en las tablas 3-7 vemos que es necesario el empleo de métodos de reordenación para reducir el llenado de los factores  $\hat{L}_i$   $\hat{U}_i$  y  $\hat{L}_S \hat{U}_S$ . Además, el empleo de estos métodos no supone empeoramiento en la velocidad de convergencia del método iterativo de Krylov.

De todos los métodos de reordenación estudiados, el que peor comportamiento ha tenido en la mayoría de las pruebas ha sido *colperm* y el que mejor *symrcm*. Aunque, hay

Matrix	size
CETAF1178	1178
CETAF5000	5021
CN2000	2038
CN3000	3020
CN5000	5005

Tabla 1: Matrices de testeo

matriz	$nnz(M)$	Tsol(s)	m	iter
CETAF1178	352346	121.37	100	2000
CETAF1178	352346	8.80	200	145
CETAF5000	322247	110.83	200	398
CN2000	202968	14.26	200	172
CN3000	211868	33.41	200	400
CN5000	256284	732.15	300	3843

Tabla 2: SPAI

casos en los que *colamd* o *symamd* funcionan un poco mejor que *symrcm*, *symrcm* es más eficiente.

Con respecto al número de partiones a realizar en los grafos de las matrices, este tiene que ser tal que el tamaño de bloque de  $\hat{A}_S$  sea inferior al del resto de bloques  $\hat{A}_i$ , ya que el trabajo de  $\hat{A}_S$  se reparte entre el número de procesadores disponibles, mientras que el trabajo de cada bloque  $\hat{A}_i$  es realizado por un único procesador. Si no se cumple esta condición no se trabaja eficientemente en un entorno paralelo.

## 4. Conclusiones

En este trabajo se ha estudiado la solución de sistemas de ecuaciones lineales derivados de diferentes aplicaciones de electromagnetismo computacional mediante un método iterativo preconditionado basado en subespacios de Krylov, GMRES(m). El preconditionador aplicado a GMRES(m) se ha obtenido como resultado de un preconditionador factorizado tipo ILU basado en técnicas de particionado de grafos. Para reducir el llenado de los

p	método	tol	$nnz(\hat{A})$	$nnz(\hat{L} + \hat{U})$	$\alpha$	Tilu(s)	Tsol(s)	iter
2		0.001	113741	350570	3.08	1.63	8.75	70
2	symamd	0.001	113741	246728	2.17	1.40	4.89	69
2	colamd	0.001	113741	228631	2.01	1.15	4.78	71
2	colperm	0.001	113741	247670	2.18	1.62	5.31	69
2	symrcm	0.001	113741	281220	2.47	1.37	5.78	70

Tabla 3: Matriz CETAF1178, Gmres(100), preconditionador de la sección 2.

p	método	tol	$nnz(\hat{A})$	$nnz(\hat{L} + \hat{U})$	$\alpha$	Tilu(s)	Tsol(s)	iter
3	symamd	0.017	273628	373247	1.36	48.64	125.46	389
3	colamd	0.017	273628	334236	1.22	48.81	119.75	384
3	colperm	0.017	273628	388791	1.42	61.29	217.69	588
3	symrcm	0.017	273628	318260	1.16	46.63	110.01	380
3		0.015	341758	718166	2.10	49.13	71.30	181
3	symamd	0.015	341758	416565	1.22	63.60	45.97	142
3	colamd	0.015	341758	385623	1.13	47.06	42.57	134
3	colperm	0.015	341758	407152	1.19	48.86	47.17	146
3	symrcm	0.015	341758	387726	1.13	44.63	43.65	141

Tabla 4: Matriz CETAF5000, Gmres(200), preconditionador de la sección 2.

p	método	tol	$nnz(\hat{A})$	$nnz(\hat{L} + \hat{U})$	$\alpha$	Tilu	Tsol	iter
2		0.0014	174114	452575	2.60	4.91	44.00	155
2	symamd	0.0014	174114	298175	1.71	3.11	26.86	153
2	colamd	0.0014	174114	339070	1.95	3.34	29.44	155
2	colperm	0.0014	174114	412085	2.34	5.34	38.84	147
2	symrcm	0.0014	174114	320000	1.84	3.03	27.29	153

Tabla 5: Matriz CN2000, Gmres(200), preconditionador de la sección 2.

p	método	tol	$nnz(\hat{A})$	$nnz(\hat{L} + \hat{U})$	$\alpha$	Tilu	Tsol	iter
2		0.0041	197010	601657	3.02	7.78	92.54	337
2	symamd	0.0041	197010	367150	1.86	6.88		†
2	colamd	0.0041	197010	425015	2.16	7.32		†
2	colperm	0.0041	197010	389905	1.97	80.74		†
2	symrcm	0.0041	197010	364830	1.85	5.88	33.36	269

Tabla 6: Matriz CN3000, Gmres(200), preconditionador de la sección 2.

p	método	tol	$nnz(\hat{A})$	$nnz(\hat{L} + \hat{U})$	$\alpha$	Tilu	Tsol	iter
2	symrcm	0.007	251887	548754	2.17	27.43	201.41	580
2	symrcm	0.01	268363	472152	1.76	26.92	386.75	1194
3	symamd	0.0125	411255	493294	1.22	31.11		†
3	colamd	0.0125	411255	618394	1.50	33.52	208.82	583
3	colperm	0.0125	411255	500305	1.22	35.12		†
3	symrcm	0.0125	411255	449996	1.09	28.44	96.60	275

Tabla 7: Matriz CN5000, Gmres(300), preconditionador de la sección 2.

factores tipo ILU ha sido necesario el empleo de matrices de reordenación, sin que ello suponga una repercusión en la velocidad de convergencia. De los métodos de reordenación estudiados, el que ha presentado mayor efectividad en líneas generales ha sido *symrcm* y el que menor *colperm*. Los resultados numéricos preliminares muestran que el preconditionador presentado en la sección 2 es competitivo frente a otras técnicas.

En un trabajo futuro se implementará el código en paralelo utilizando el lenguaje de programación fortran 90 junto con librerías MPI.

## Referencias

- [1] M. Benzi and J. Marín and M. Tuma. A Two-Level Parallel Preconditioner Based on Sparse Approximate Inverses. *Iterative Methods in Scientific Computation IV*, 5:167–177, 1999.
- [2] B. Carpentieri, I. S. Duff, and L. Giraud. Sparse pattern selection strategies for robust Frobenius-norm minimization preconditioners in electromagnetism. *Numerical Linear Algebra with Applications*, 7(7-8):667–685, 2000.
- [3] B. Carpentieri. *Sparse preconditioners for dense linear systems, from electromagnetics applications*. PhD thesis, l’Institut National Polytechnique de Toulouse, CERFACS, 2002.
- [4] B. Carpentieri. *A class of spectral two-level preconditioners*. *SIAM J. Sci. Comput.*, 25(2):749–765, 2003.
- [5] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *Journal of Computational Physics*, 73(3):325–348, 1987.
- [6] M. Grote and T. Huckle. Parallel preconditioning with sparse approximate inverses. *SIAM Journal on Scientific Computing*, 18(3):838–853, 1997.
- [7] R. Harrington. Origin and development of the Method of Moments for field computation. *IEEE Antennas and Propagation Magazine*, 1990.
- [8] G. Karypis, and V. Kumar. : A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.*, 20:359–392, 1999.
- [9] K.S. Kunz and R.J. Luebbers. The Finite Difference Time Domain Method for Electromagnetics. *SIAM J. Sci. Comput.*, 18(3):838–853, 1997.
- [10] D. C. Sorensen R. B. Lehoucq and C. Yang. *ARPACK Users’ Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*. SIAM, Philadelphia, 1998.
- [11] T. A. Manteuffel. An incomplete factorization technique for positive definite linear systems. *Math. Comp.*, 34:473–497, 1980.
- [12] N. Malla and J. Marín and E. Pascual. Spectral Low Rank Preconditioning for Computational Electromagnetics Applications. *Proceedings of The Sixth International Conference on Engineering Computational Technology*, Edited by M. Papadrakakis and B.H.V. Topping, Athens, Greece, 2-5 September, 2008.
- [13] J. C. Maxwell. A dynamical theory of the electromagnetic field. *Royal Society Transactions*, CLV, 1864. Reprinted in R. A. R. Tricker, *The Contributions of Faraday and Maxwell to Electrical Science*, (Pergamon Press, 1966).
- [14] J. A. Meijerink and H. A. van der Vorst. An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. *Math. Comp.*, 31:148–162, 1977.
- [15] S. M. Rao and D. R. Wilton and A. W. Glisson. Electromagnetic scattering by surfaces of arbitrary shape. *IEEE Trans. Antennas Propagat.*, AP-30:409–418, 1982.
- [16] Y. Saad. ILUT: a dual threshold incomplete ILU factorization. *Numerical Linear Algebra with Applications*, 1:387–402, 1994.
- [17] Y. Saad. *Iterative Methods for Sparse Linear Systems*. PWS Publishing Company, Boston, 1996.
- [18] P. P. Silvester and R. L. Ferrari. *Finite Elements for Electrical Engineers*. Cambridge University Press, Cambridge, 1990.
- [19] H. A. van der Vorst. Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of non-symmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 12(6):631–644, 1992.